

Project HTSM

E

Ontwerp document

Student nummer: 366287

Versie 05052025

Pas TAN van der, Tim

Inhoudsopgave

1	Inleiding	1
1.1	Opdracht omschrijving:	1
1.2	Methodiek	1
2	Uitwerking van opdracht	2
2.1	Aanwezige apparaten	2
2.2	Scenario	4
2.3	Functionele specificaties	7
2.4	Hoofd algoritme	11
2.5	Specificaties	12
2.5.1	Hoofdalgorimte	12
2.5.2	Deel algoritmes	12
2.6	Samenvatting	16
3	Plan van aanpak realisatie en testen	17
3.1	Planning	19
3.2	Testen	20
4	Referentielijst	21
	Figuren	21
	Tabellen	21
	Bijlagen	21
	Referentielijst	21
	Bijlagen	22

1 Inleiding

Dit hoofdstuk beschrijft de aanpak voor de realisatie van de besturingsapplicatie die het huishoudelijke elektriciteitsnetwerk in balans houdt. De applicatie monitort de accusatus en regelt het aan- en uitschakelen van apparaten om een stabiele energievoorziening te garanderen. De implementatie omvat het schrijven van het besturingsalgoritme in de programmeertaal C, waarbij gedetailleerde specificaties vanuit eerdere analyses en ontwerpen worden vertaald naar werkende programmatuur.

1.1 Opdracht omschrijving:

De projectopdracht behelst het schrijven van een besturingsapplicatie die het ‘huis-elektriciteitsnetwerk’ in balans houdt door de status van de accu te monitoren en bepaalde verbruiksapparaten actief aan en/of uit te schakelen.

Het project is opgedeeld in 3 stappen:

1. Analyse en ontwerp van een besturingsalgoritme, opleverdocument output dit ontwerp document.
2. Realisatie daarvan in de taal C, opleverdocument code in vorm van een cbp. bestand
3. Oplevering, opleverdocument video.

1.2 Methodiek

De methodiek voor deze realisatiefase is gebaseerd op het gebruik van Flowgorithm, waarmee heldere en gedetailleerde flowcharts zijn ontwikkeld. Deze flowcharts worden vervolgens als basis gebruikt voor de praktische implementatie in C. Specifiek wordt aandacht besteed aan het structureren van programma-onderdelen in duidelijke functies en datastructuren, zoals de structuur 'last', en het uitvoeren van uitgebreide systeemtesten volgens vooraf bepaalde scenario's om betrouwbaarheid en correctheid van de applicatie te verzekeren. Hierna zal moet codeblocks een programma geschreven worden waarmee de functies zoals uitgewerkt in Flowgorithm worden vertaald in C code. Deze code zal vervolgens in een video worden toegelicht om het vak te kunnen afsluiten.

2 Uitwerking van opdracht

2.1 Aanwezige apparaten

Zie Tabel 1 Overzicht van gebruikers en afgevers energie hier is een opsomming gegeven van de opwekkers en verbruikers en het vermogen van het betreffende apparaat.

Tabel 1 Overzicht van gebruikers en afgevers energie

Apparaat	Vermogen in watt	Opwekker/gebruiker	Prioriteit bij uitschakelen	Afbeelding
Gloeilamp 1	25	gebruiker	10	
Gloeilamp 2	40	gebruiker	9	
Gloeilamp 3	60	gebruiker	8	
Gloeilamp 4	100	gebruiker	6	
Gloeilamp 5	150	gebruiker	5	
Plasma-TV	380	gebruiker	3	
Airco	850	gebruiker	1	
Ijskast	75	gebruiker	7	

Waterkoker	800	gebruiker	2	
PC	250	gebruiker	4	
Dumpload	0 – 3000	gebruiker		
Accu	0 -300 opladen 0 – 1500 afgeven	Opwekker/gebruiker		
Windmolen	0 - 2500	Opwekker		
Zonnecellen	0 – 300	Opwekker		

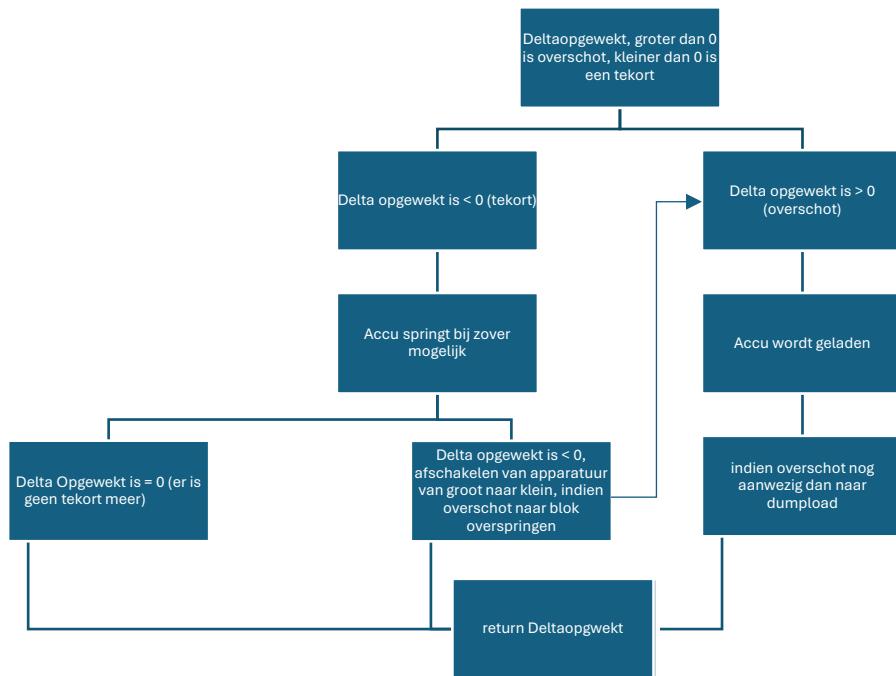
2.2 Scenario

Om een idee te hebben we de in en output van het systeem zouden kunnen zijn. Zijn een aantal scenario's uitgewerkt. Dit helpt bij het verder uitwerken van de programmatuur en het testen om te controleren of het systeem aan de eisen voldoet.

Tabel 2 Scenario's van systeem

	Opgewekt e vermogen	Verbuikt vermogen	Delta	accu	Dumplod ad	Afschakelen
Max wind en Zon, en max verbruikt	2800	2730	70	70	0	n.v.t.
Max wind en zon geen verbruik, geen accu opladen	2800	0	2800	0	2800	n.v.t.
Max wind en zon geen verbruik, accu opladen	2800	0	2800	300	2500	n.v.t.
Max wind, max verbruik, geen accu	2500	2730	230	0	0	230 (airco)
Max zon, max verbruik, geen accu	300	2730	2430	0	0	Airco Waterkoker Plasma-TV PC Gloeilamp 5
50 % Wind % 50 zon, max verbruik	1400	2730	1330	0	0	Airco Waterkoker
Geen zon en wind alleen accu, max verbruik	300 (accu)	2730	2430	0	0	Airco Waterkoker Plasma-TV PC Gloeilamp 5
Wind kleiinder dan 0 of groter dan 2500	2500	n.v.t.				Wind minimalseren op 0, en maximaliseren op 2500
Zon kleiinder dan 0 of groter dan 300	300	n.v.t.				Zon minimaliseren op 0 en

						maximaliseren op 300
--	--	--	--	--	--	----------------------



Figuur 1 Code voor deel van het bepalen tekort/overschat

Tabel 3 Eisen aan ontwerp algemeen apparaten

Eis	Methode	Actie	Flow diagram
Een zestal zonnepanelen: Deze leveren samen een elektrisch vermogen van maximaal 300W	Maximaliseren van output van panelen op 300 W	1st run laden van beginwaarde,	
Een windmolen: Deze levert een elektrisch vermogen van maximaal 2.5kW	Maximaliseren van output op 2.5Kw.	Maximaliseren van output op 2.5Kw.	
Een accu-batterij: Deze kan een maximale hoeveelheid vermogen leveren van 1.5kW, en de accu batterij kan maximaal met 300W opgeladen worden	Max 1.5 Kw leveren en 300w opgeladen.	Maximaliseren 1.5 Kw leveren en 300w opgeladen.	
De belastingen: De volgende belastingen zijn aangesloten op het systeem: 5 gloeilampen (van 25W, 40W,	Belastingen toevoegen.	Waarde van apparaat aanpassen.	

60W, 100W en 150W), 1 plasma-TV van 380W, 1 airco van 850W, 1 ijskast van 75W, 1 waterkoker van 800W en 1 computer van 250W			
Een 3kW-dump load. Deze load wordt ingeschakeld als er wel vermogen opgewekt wordt terwijl er geen vraag is naar stroom. Dit is met name voor de windmolen van belang: deze zou anders bij een sterke wind zijn energie niet kwijt kunnen en op hol kunnen slaan. Het vermogen dat de dump load dissipeert is instelbaar van 0-3kW.	Dumupload toevoegen	Programma aanpassen dat bij een overschot alles naar de dumupload gaat. Dumupload krijgt dan de waarde van het overschot.	

2.3 Functionele specificaties

Tabel 4 Eisen functionele specificaties

Eis	Methode
	Flow diagram
Functionele specificaties	
De besturing zorgt er ten alle tijden voor dat de wet van behoud van energie klopt: dat wil zeggen dat de som van het opgewekte vermogen altijd gelijk is aan de som van het afgenomen vermogen.	Berekening van opgewekt vermogen, berekening van geconsumeerd vermogen, terug schakelen van geconsumeerd vermogen, bij tekort, bij overschot gaat vermogen naar dumpload
De besturing zorgt ervoor dat de beschikbare energie optimaal benut wordt. Hiertoe moet één van de prioritering systemen gekozen worden die in bijlage III beschreven zijn. Bij een verandering in het energie-aanbod besluit het besturingssysteem welke last in- of uitgeschakeld wordt op basis van het prioritering systeem	Apparaten uitwisschakelen totdat het tekort nul is.
De accu-batterij werkt als een energie-buffer: als de windmolen en/of zonnepanelen meer vermogen opwekken dan er op dat moment verbruikt wordt, wordt dit verschil in de accu opgeslagen tot een maximum van 300W. De energie van de accu wordt gebruikt als er niet voldoende energie opgewekt wordt. Als het vermogen surplus meer is dan 300W, of als de accu vol is, dan wordt het resterende overschot (na aftrek mogelijk accu laad vermogen) in de dumpload gedissipeerd.	Berekenen wat het beschikbare overschot is als dit minder dan 300 W is gaat dit naar de accu.
Om de accu te sparen geldt het volgende voor het gebruik van de accu: <ul style="list-style-type: none"> - Bij een lading van minder dan 20% zal de accu nooit vermogen leveren (anders gaat hij stuk) - Bij een lading tussen de 20% en 39% is het maximale vermogen gelijk aan 300W - Bij een lading tussen de 40% en 59% is het maximale vermogen gelijk aan 600W - Bij een lading tussen de 60% en 79% is het maximale vermogen gelijk aan 1.0kW - Bij een lading tussen de 80% en 100% is het maximale vermogen gelijk aan 1.5kW 	Vergelijking maken met de waarde van de accu, en hetgeen teruggeleverd wordt.

- Bij een lading van 100% is geen laadvermogen toegestaan	
Door het drukken op de aan-schakelaar van een last geeft de gebruiker aan dat hij de last wilt inschakelen. Het besturingssysteem bepaalt aan de hand van het prioriteringsysteem en het op dat moment beschikbare vermogen of de last ook daadwerkelijk ingeschakeld wordt. De prioritering kan ertoe leiden dat het inschakelen van de ene last resulteert in het uitschakelen van een andere last.	Berekenen opgenomen en opgewekt van mogen, tekort bepalen en basis daarvan volgens de prioritering apparaten afschakelen totdat tekort 0 is.
Door het drukken op de uit-schakelaar van een last geeft de gebruiker aan dat hij de last wilt uitschakelen: deze handeling wordt altijd door het besturingssysteem uitgevoerd.	Waarde van last aanpassen naar uitgeschakeld, Overschot bepalen en naar dumpload sturen.
Overige requirements	
Om het echte elektriciteits-prosumer-systeem met een PC te kunnen simuleren, zijn de volgende gebruikersinterfaces opgesteld: o <i>De invoer-interface</i> Bij een echt elektriciteits-prosumer-systeem zal de invoer in ieder geval bestaan uit : - sensoren, die aangeven hoeveel vermogen de windmolen en zonnepanelen leveren - een sensor die aangeeft hoeveel lading de accu heeft - schakelaars waarmee de gebruiker de lasten aan- en uit kan schakelen.	Waardes van vermogen windmalen en zonnepalen, waarde accu, status schakelaars printen naar scherm.
Bij de PC-applicatie voert de gebruiker deze gegevens in door commando's (zie Tabel 1 in de appendix I) in te geven via het toetsenbord of <i>via een op de commandoregel op te geven invoerbestand met daarin commando's</i> . In zo'n bestand is er maximaal één commando per regel en elk commando moet uitgevoerd worden tot het eind van het bestand of het commando 'invoer keyboard' .	Commando controleren, waarde op basis van vergelijking aanpassen.
Bij een echt elektriciteits-prosumer-systeem zal de uitvoer bestaan uit: - aansturingen voor de relais-schakelaars die de lasten fysiek in en uitschakelen	Uitvoer printen naar scherm.

<ul style="list-style-type: none"> - een aansturing voor het accu-vermogen - een aansturing voor de dump load - eventueel een display met additionele informatie. <p>In de PC-applicatie bestaat de uitvoer uit (zie ook Figuur 2 in appendix I):</p> <ul style="list-style-type: none"> - het vermogen dat de windmolen levert - het vermogen dat de zonnepanelen leveren. - het vermogen dat de accu levert of opneemt (gebruik een positief teken als de accu energie levert en een negatief teken als deze energie opneemt) - het vermogen dat de dumpload dissipeert - een duidelijk overzicht van de stand van de relais-schakelaars van alle lasten - de output wordt standaard op het scherm getoond. In geval dat op de commandoregel gespecificeerd is in de opgegeven output file geschreven, waarbij de namen gebruikt worden zoals in tabel 2, net zolang tot er een commando ‘uitvoer scherm’ volgt. Alle apparaten in een formaat met vaste kolommen en gescheiden door komma’s op een enkele regel. 	
<p>Bij het starten van de PC-applicatie wordt uitgegaan van vooraf gedefinieerde begintoestand. Zie Tabel 2 in de appendix I voor de ‘standaard’ waarden. De begintoestand data moet uit een begintoestand.txt bestand lezen worden.</p>	<p>Laden van tekst bestand, waardes opslaan bij juiste variabelen.</p>
Appendix II. Programmeervereisten en tips	
<p>het programma moet correct werken (logisch ...)</p>	<p>Testen op basis scenario’s.</p>
<p>het programma moet een goede layout en een overzichtelijke structuur hebben, relevant en zinvol commentaar en naamgeving van functies en variabelen.</p>	<p>Controle op leesbaarheid en opmerkingen.</p>
<p>Het programma moet correcte en veilige code hebben, vooral op plaatsen waar gemakkelijk iets fout kan gaan (scannen van commando’s of argumenten, file IO). Zowel</p>	<p>Testen op basis scenario’s.</p>

2.4 Hoofd algoritme

Om een idee te krijgen hoe het hoofdalgoritme er uit kan zien. Is een eerste concept gemaakt van deze opbouw. Zie daarvoor de onderstaande tabel.

Tabel 5 Lautout van code concept

Stap	Omschrijving	
1	Declareren van variabelen (vermogen, status aan of uit, functies, bibliotheken, Tekst bestand)	Benodigde variabelen:
	Main	
2	Inlezen start status waarden variabelen vanuit tekst bestand (alleen de eerste keer)	
3	Waarde status van verbruikers, opwekkers aanpassen n.a.v. commando	Overschrijven van waarde, die initieel uit tekstbestand gehaald is.
4	Berekenen opgewekt vermogen	Check of opwekker aan staat, sommatie hiervan
5	Berekenen verbruikt vermogen	Check of verbruiker aanstaat. sommatie hiervan
6	Berekenen van tekort/overschot vermogen	Opgewekt vermogen – verbruikt vermogen.
7	D.m.v. prioritering vermogen berekening maken wat nodig is afschalen tot verbruikt vermogen < opgewekt vermogen.	Van overschot- verbruikers aftrekken tot waarden kleiner is dan 0.
8	Status van verbruikers aanpassen a.d.h.v. berekening overschot.	
9	Indien nodig, delta van opgewekt vs verbruikt vermogen naar dumpload sturen.	
10	Print van status van opwekkers en verbruikers	
11	Print totalen van verbruikt en opgewekt.	
12	Print vragen om commando's	
13	Commando opslaan.	
14	Print van Commando's lijst. (indien gevraagd)	
	Functies	
	D.m.v. prioritering vermogen berekening maken wat nodig is afschalen tot verbruikt vermogen < opgewekt vermogen.	

2.5 Specificaties

In dit hoofdstuk worden verschillende algoritmes besproken, hoe ze zijn opgebouwd en wat hun functie is.

2.5.1 Hoofdalgoritmte

Zie Bijlage 1 Main voor de flow van het programma. Dit dient nog verder in codeblocks te worden aangepast. Echter zijn de mogelijkheden in flowgorithm beperkt om het programma zo slank mogelijk te krijgen. Allereerst zijn er aantal variabelen gedeclareerd. Dit kan niet in een aparte functie in flowgorithm.

2.5.2 Deel algoritmes

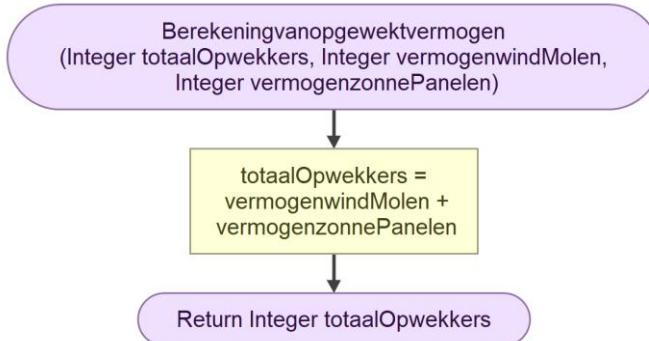
Commando

In het commando algoritme wordt gecontroleerd wat er getypt is en met welk commando dit overeen komt en basis daarvan wordt een waarde bijv. een lamp aan of uit gezet. Zie Bijlage 6 Commando voor de uitwerking hiervan in flowgorithm.

Algoritme instellen zon, wind en accu

Dit algoritme zorgt ervoor dat de gebruiker een instelling kan aanpassen (voor windvermogen, zonnevermogen of acculading) en waarborgt dat de ingevoerde waarde altijd binnen vooraf bepaalde maximale grenzen blijft. Hiermee voorkomt het algoritme dat onrealistische waarden ingesteld kunnen worden voor de systemen waarmee het algoritme werkt (windmolen, zonnepanelen of accu). Zie Bijlage 2 InstellenWindZondAccu voor flow in flowgorithm.

Berekening van opgewekt vermogen

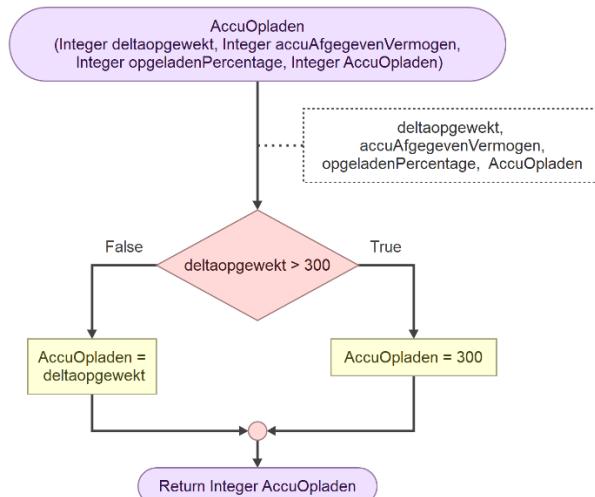


Figuur 2 berekening van opgewekt vermogen

Algoritme berekening van verbruikt vermogen

Zie Bijlage 5 Berekening van verbruikt vermogen voor de uitwerking van flowgorithm.

Algoritme voor het bepalen opladen accu.



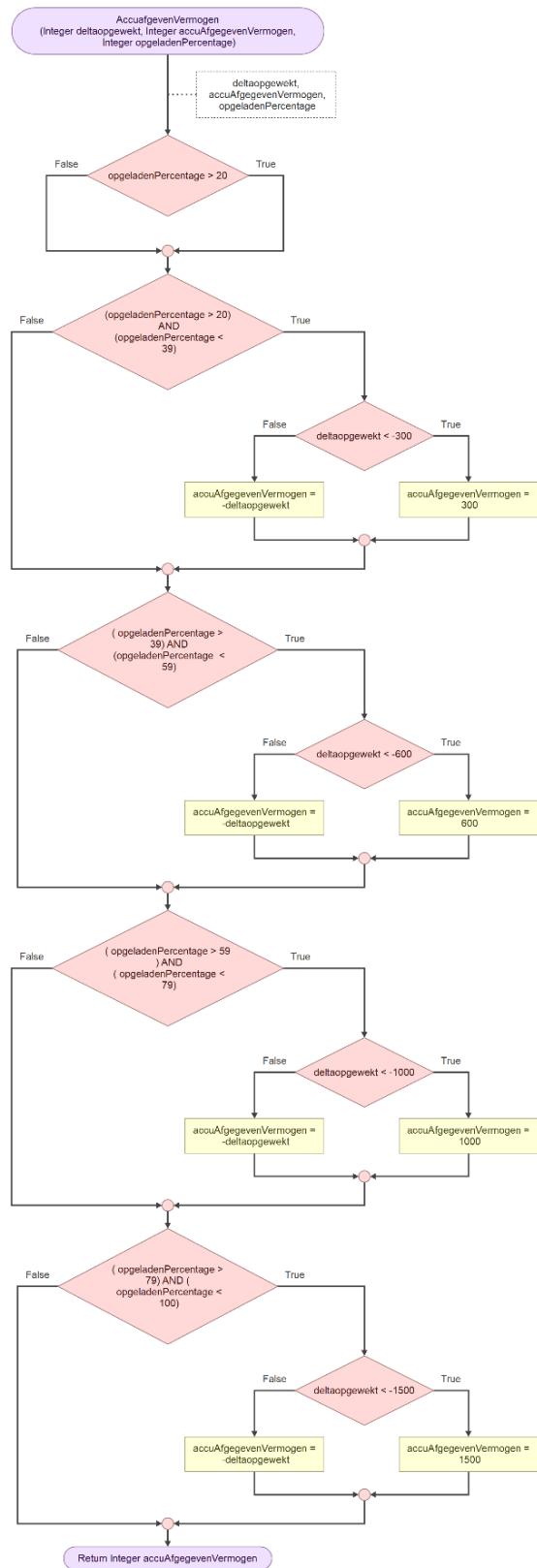
Figuur 3 Algortome opladen accu

Algoritme prioritering van uit te schakelen apparaten.

Er is gekozen voor Prioriteringsysteem A, dit is naar verwachting het meest eenvoudig te implementeren in de code. Zie Bijlage 4 Berekening van uitschakelen apparaten voor de uitwerking in flowgorithm.

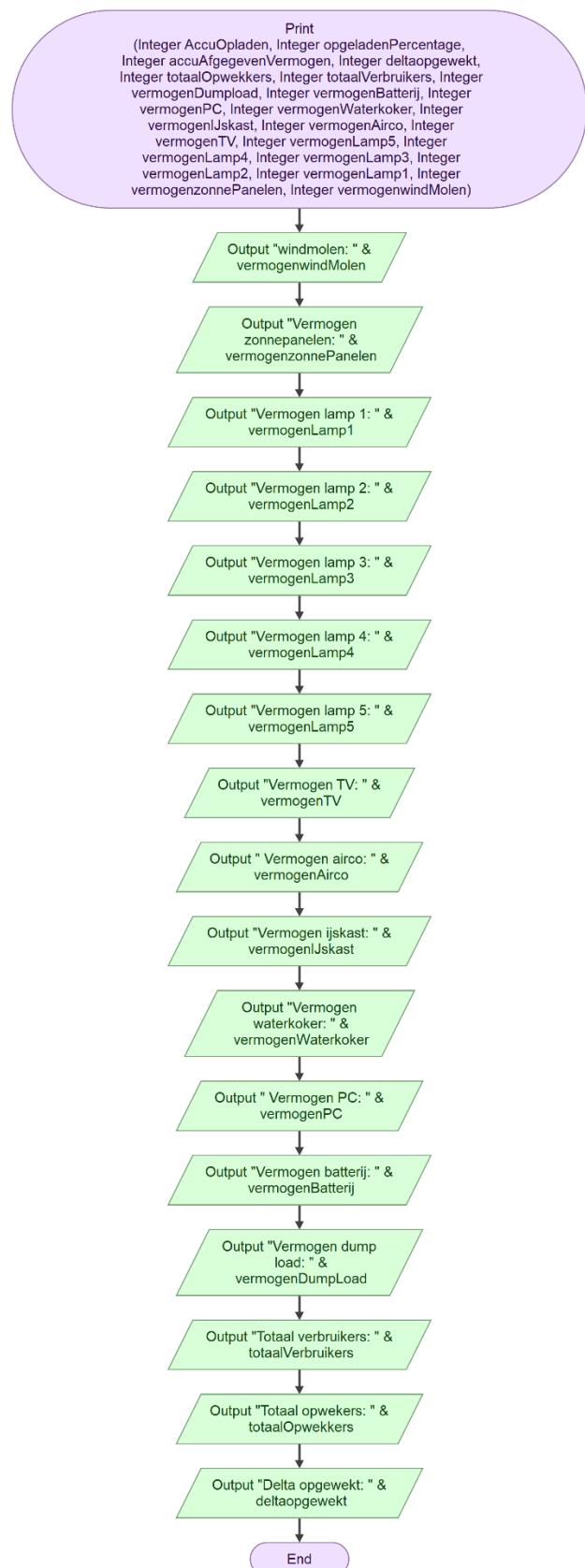
Algoritme voor het bepalen van afgeven vermogen accu.

Dit algoritme beheert het energieverbruik van de diverse apparaten door rekening te houden met het beschikbare vermogen. Het controleert in een vaste volgorde per apparaat of er voldoende vermogen over is om het apparaat in te schakelen. Als er niet genoeg vermogen beschikbaar is, wordt het apparaat uitgeschakeld, het resterend beschikbare vermogen verminderd met het verbruik van het apparaat, en de status van het apparaat op nonactief gezet. Als er niet genoeg vermogen is, blijft het apparaat uitgeschakeld en gaat het algoritme door naar het volgende apparaat. Uiteindelijk retourneert het algoritme de resterende hoeveelheid vermogen die nog over is na het in- of uitschakelen van alle apparaten.



Figuur 4 Flow accuvermogen

Print algoritme hiermee worden de verschillende waarde van het programma geprint.



Figuur 5 Flow print code

2.6 Samenvatting

De functie Berekeningvanopgewektvermogen(Integer totaalOpwekkers, Integer vermogenwindMolen, Integer vermogenzonnenPanelen) berekent het totaal opgewekte vermogen uit zonne- en windenergie. De functie InstellenWindZondAccu(Integer werkelijkvermogenwindMolen, Integer werkelijkvermogenzonnenPanelen, Integer werkelijkopgeladenPercentage, String inputkeyboard2) stelt gebruikers in staat om de waarden voor windenergie, zonne-energie en acculading binnen gespecificeerde limieten in te stellen. Commando(String inputkeyboard, Boolean statusLamp1 t/m Lamp5, statusTV, statusAirco, statusIjskast, statusWaterkoker, statusPC, Integer vermogenLamp1 t/m Lamp3, werkelijkvermogenLamp1 t/m Lamp3) verwerkt gebruikerscommando's voor het schakelen van apparaten. Print(Integer AccuOpladen, Integer opgeladenPercentage, Integer accuAfgegevenVermogen, Integer deltaopgewekt, Integer totaalOpwekkers, Integer totaalVerbruikers, Integer vermogenDumpLoad, Integer vermogens per apparaat) verzorgt het overzichtelijk weergeven van alle vermogensstatussen op het scherm. De functie Berekeningvanuiteschakelenapparaten(Integer deltaopgewekt, Boolean statusLamp2 t/m Lamp5, statusAirco, statusTV, statusIjskast, statusWaterkoker, statusPC, Integer vermogens) bepaalt welke apparaten uitgeschakeld moeten worden op basis van prioriteit bij energietekort. Daarnaast regelen functies als AccuOpladen(Integer deltaopgewekt, Integer accuAfgegevenVermogen, Integer opgeladenPercentage, Integer AccuOpladen) en AccuafgevenVermogen(Integer deltaopgewekt, Integer accuAfgegevenVermogen, Integer opgeladenPercentage) het laden en ontladen van de accu volgens vastgelegde regels voor acculading. Tot slot bepaalt Berekeningvanverbruiktvermogen(Integer vermogens, Boolean status per apparaat) het totale energieverbruik door alle actieve apparaten op te tellen

3 Plan van aanpak realisatie en testen.

Dit hoofdstuk beschrijft de realisatie en implementatie van het besturingssysteem voor het huishoudelijke elektriciteitsnetwerk. Het begint met het initialiseren van specifieke variabelen en datastructuren, zoals de structuur 'last' voor elk apparaat, en het voorbereiden van het bestand 'begintoestand.txt'. Vervolgens wordt het hoofdalgoritme geïmplementeerd, waarbij flowcharts worden vertaald naar programmeerfuncties voor invoerverwerking, energieproductie- en verbruksberekeningen, en accubeheer op basis van prioriteitsregels. Tot slot bespreekt het hoofdstuk modulaire integratie en uitgebreide systeemtesten om correcte werking onder uiteenlopende scenario's te garanderen.

1. Initialiseren van specifieke variabelen en datastructuren:

- Aanmaken en initialiseren van project-specifieke datastructuren, zoals struct "apparaat", welke voor elk apparaat de velden naam, vermogen, prioriteitspunten, aan en uit.
- Voorbereiden van het bestand "begintoestand.txt" met de standaardwaarden uit tabel 2 voor dit specifieke huishoudelijke netwerk.
- Variabelen, Functies, structures, declareren

2. Implementeren van het hoofdalgoritme:

- Implementatie van het gedetailleerde hoofdalgoritme zoals uitgewerkt in het specifieke flowchart. Alleen functies onder elkaar vermelden, met opmerkingen de rol van elke functie.
- Vertalen van flowcharts naar expliciete functies, specifiek gericht op:
 - Lezen van invoer van toetsenbord
 - Variabelen updaten met de invoer van toetsenbord.
 - Het berekenen van vermogen geproduceerd door zonnepanelen en windmolen.
 - Het berekenen van totaal energieverbruik van specifieke apparaten.
 - Op basis van tekort een functie aanroepen voor het afschakelen, danwel een functie aanroepen voor het opladen van de accu.
 1. Het regelen van laad- en ontlaadprocessen van de accu.
 2. Uitvoeren van de prioriteitsberekeningen

- Print van waardes apparaten, opgewekte vermogens en accu.

3. Modulaire integratie en testen van componenten:

- Afzonderlijk implementeren en testen van iedere module (zoals accubeleer, apparatenbeheer en commandoverwerking)
- Uitvoeren van uitgebreide integratietesten om de interactie tussen modules te valideren.

4. Uitvoeren van gedetailleerde systeemtesten:

- Uitgebreide testen volgens vooraf bepaalde scenario's uit Tabel 2 Scenario's van systeem.

3.1 Planning

In de onderstaande tabel is een planning te vinden van de werkzaamheden. Deze planning is een leidraad van welke activiteiten en welke volgorde uitgevoerd zullen worden.

Tabel 6 Planning

Fase	Taken	Startdatum	Einddatum	Mijlpalen	Document out
Voorbereiding & Initialisatie	Opzetten datastructuren, lezen begintoestand, variabelen declareren	6-5-2025	12-5-2025	Datastructuren gereed, begintoestand geladen	code
Implementatie Hoofdalgoritme	Implementeren hoofdflow (energiebeheer, inputverwerking)	13-5-2025	19-5-2025	Werkend hoofdalgoritme	code
Implementatie Deelalgoritmes	Implementatie specifieke functies (prioritering, accubeheer, commandoverwerking)	20-5-2025	26-5-2025	Alle deelalgoritmes geïmplementeerd	code
Modulaire Integratie	Integratie afzonderlijke modules, oplossen integratieproblemen	20-5-2025	26-5-2025	Succesvolle integratietest	code
Uitgebreide Testfase	Uitvoeren van tests volgens scenario's, verwerken feedback	3-6-2025	9-6-2025	Voltooiing testscenario's	code in bestand.cbp
Optimalisatie & Afronding	Code-optimalisatie, documentatie afronden en definitieve oplevering dmv video	10-6-2025	16-6-2025	Afronding optimalisatie en documentatie	Video

3.2 Testen

Zie de onderstaande tabel voor de testen en de uitkomsten die het systeem moet geven.

Tabel 7 Testen en uitkomsten

cenario	Opgewekt Vermogen (W)	Verbruikt Vermogen (W)	Verwachte Uitkomst	Succescriteria
Max wind en zon, max verbruik	2800	2730	Alle apparaten aan, overshot naar accu	Geen apparaten uitgeschakeld, accu laadt correct overshot
Max wind en zon, geen verbruik, geen accu opladen	2800	0	Volledig overshot naar dumpload	Geen energie gebruikt, dumpload krijgt volledige overshot
Max wind en zon, geen verbruik, accu opladen	2800	0	Accu opladen tot limiet, rest naar dumpload	Accu opladen maximaal 300W, rest overshot correct naar dumpload
Max wind, max verbruik, geen accu	2500	2730	Airco uitgeschakeld (hoogste verbruiker)	Airco correct uitgeschakeld bij energietekort
Max zon, max verbruik, geen accu	300	2730	Airco, waterkoker, plasma-TV, PC, Gloeilamp 5 uitgeschakeld	Alle gespecificeerde apparaten correct uitgeschakeld volgens prioriteit
50% wind, 50% zon, max verbruik	1400	2730	Airco en waterkoker uitgeschakeld	Airco en waterkoker correct uitgeschakeld volgens prioriteit
Geen zon en wind, alleen accu, max verbruik	300 (accu)	2730	Airco, waterkoker, plasma-TV, PC, Gloeilamp 5 uitgeschakeld, accu gebruikt	Accu correct gebruikt, gespecificeerde apparaten correct uitgeschakeld

4 Referentielijst

Figuren

Figuur 1 Code voor deel van het bepalen tekort/overschot.....	5
Figuur 2 berekening van opgewekt vermogen.....	12
Figuur 3 Algortome opladen accu	13
Figuur 4 Flow accuvermogen.....	14
Figuur 5 Flow print code	15

Tabellen

Tabel 1Overzicht van gebruikers en afgevers energie	2
Tabel 2 Scenario's van systeem	4
Tabel 3 Eisen aan ontwerp algemeen apparaten	5
Tabel 4 Eisen functionele specificaties	7
Tabel 5 Lautout van code concept	11
Tabel 6 Planning	19
Tabel 7 Testen en uitkomsten	20

Bijlagen

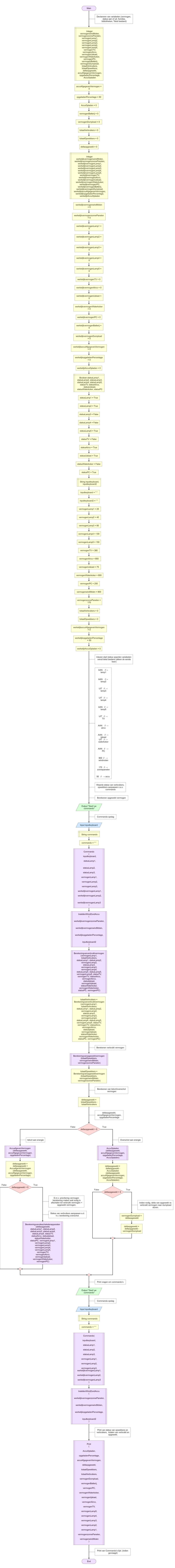
Bijlage 1 Main	23
Bijlage 2 InstellenWindZondAccu	24
Bijlage 3 AccuafgevenVermogen.....	25
Bijlage 4 Berekeningvanuiteschakelenapparaten.....	26
Bijlage 5 Berekeningvanverbruiktvermogen	27
Bijlage 6 Commando	28

Referentielijst

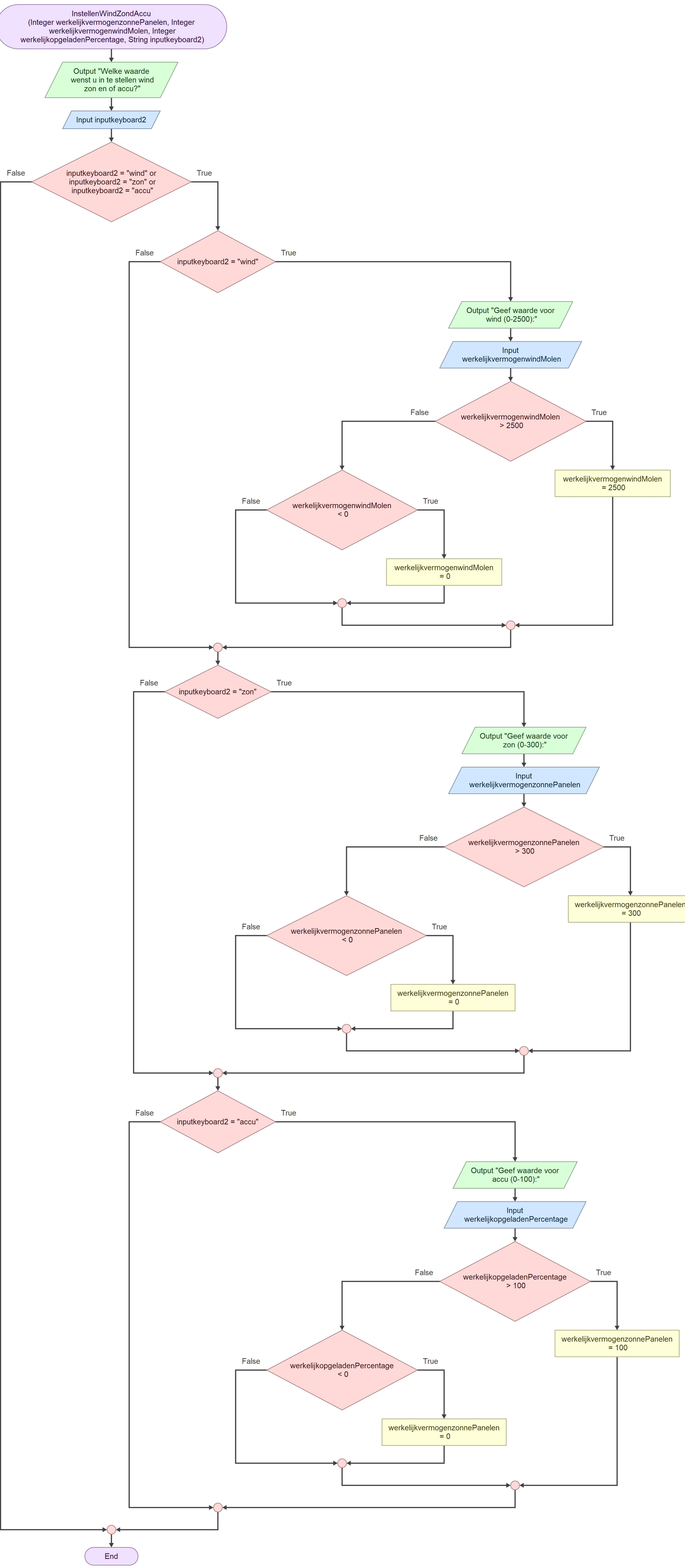
Chatgtp

Bijlagen

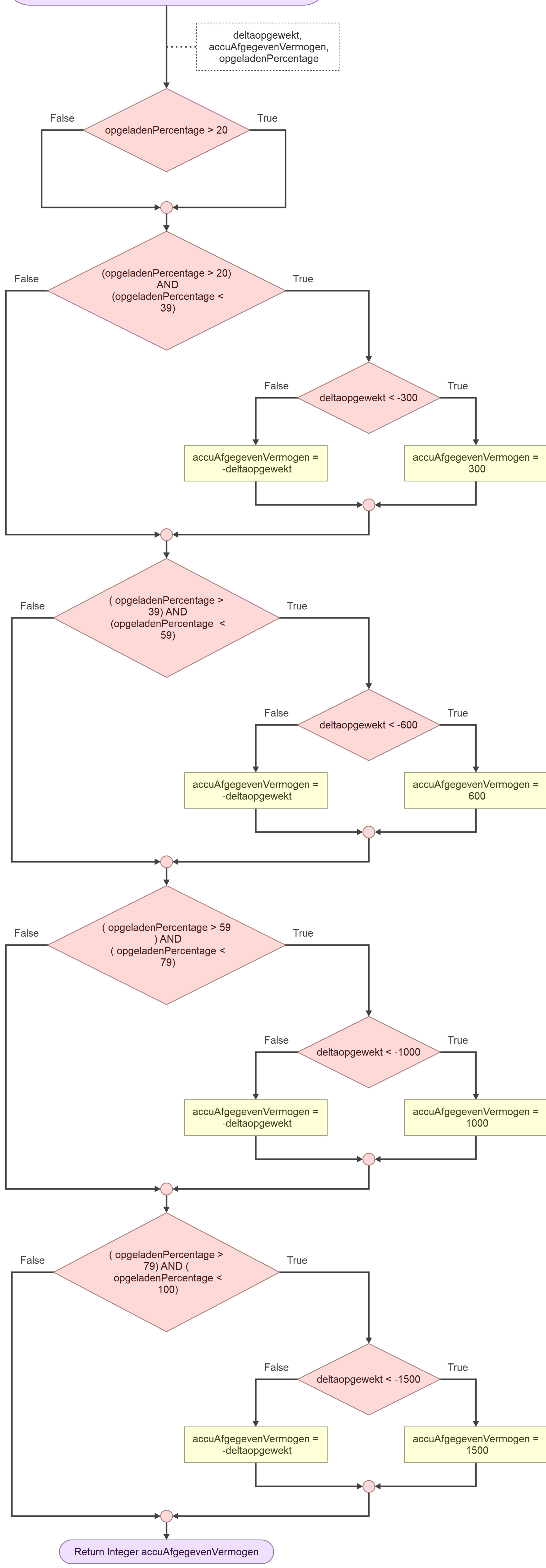
Bijlage 1 Main



Bijlage 2 InstellenWindZondAccu

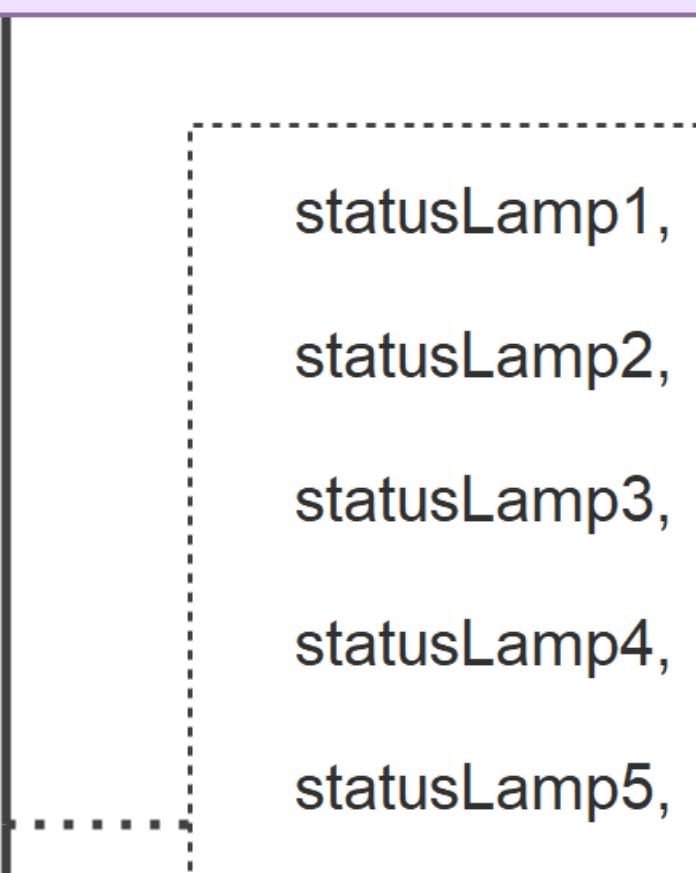


AccuAfgevenVermogen
(Integer deltaopgewekt, Integer accuAfgegevenVermogen,
Integer opgeladenPercentage)

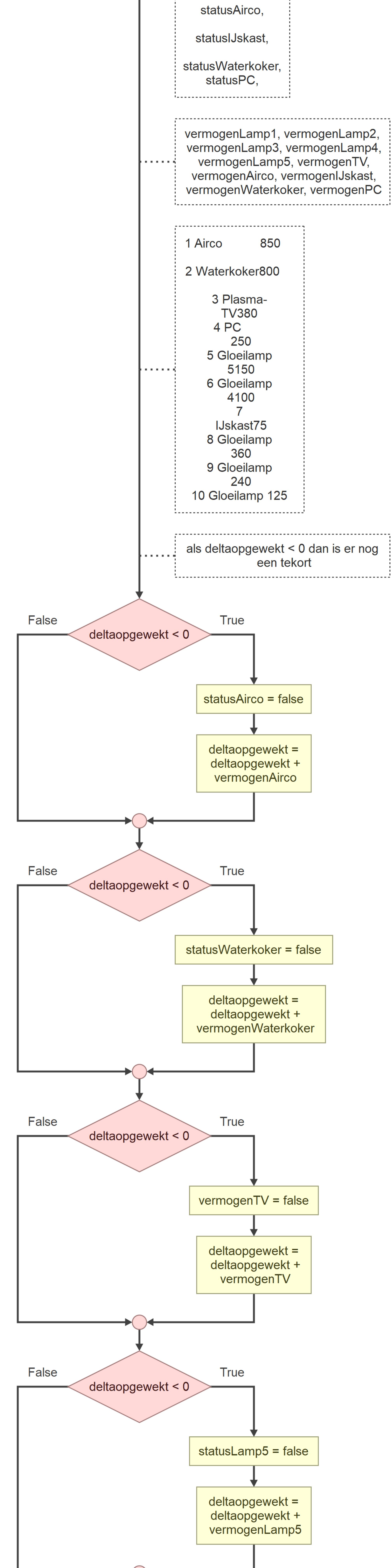


Bijlage 4 Berekening van uiteschakelen apparaten

Berekening van uiteschakelen apparaten
(Integer deltaopgewekt, Boolean statusLamp1, Boolean statusLamp2, Boolean statusWerkkoker, Boolean statusIJska, Boolean statusAirco, Boolean statusTV, Boolean statusLamp3, Boolean statusPC, Boolean statusLamp4, Boolean statusLamp5, Integer vermogenLamp1, Integer vermogenLamp2, Integer vermogenLamp3, Integer vermogenLamp4, Integer vermogenLamp5, Integer vermogenTV, Integer vermogenAirco, Integer vermogenIJska, Integer vermogenWerkkoker, Integer vermogenPC)

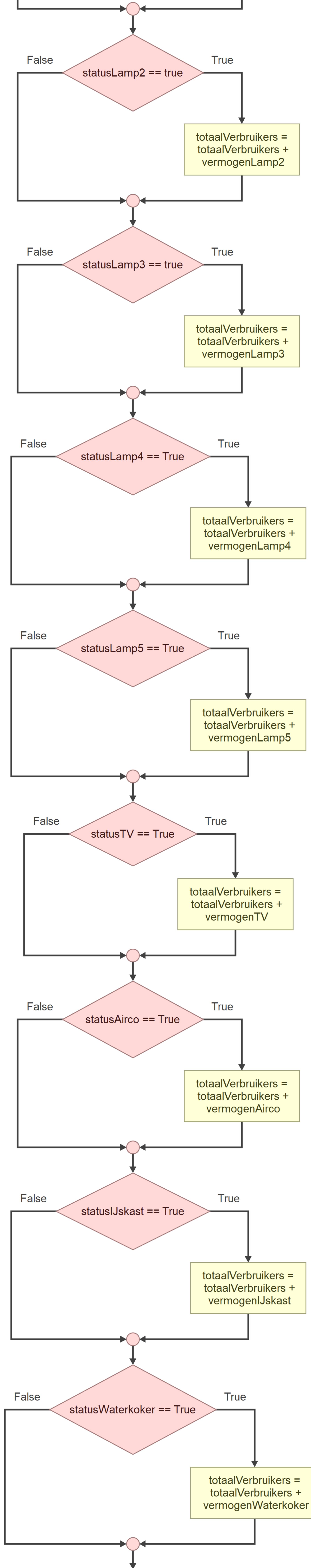


statusLamp1,
statusLamp2,
statusLamp3,
statusLamp4,
statusLamp5,



Bijlage 5 Berekening van verbruiktvermogen

Berekening van verbruiktvermogen
 (Integer vermogenLamp1, Integer totaalVerbruikers,
 Boolean statusLamp1, Boolean statusLamp2, Integer
 vermogenLamp2, Boolean statusLamp3, Integer
 vermogenLamp3, Integer vermogenLamp4, Boolean
 statusLamp4, Boolean statusLamp5, Integer
 vermogenLamp5, Boolean statusTV, Integer vermogenTV,
 Boolean statusAirco, Integer vermogenAirco, Boolean
 statusIjskast, Integer vermogenIjskast, Boolean
 statusWaterkoker, Integer vermogenWaterkoker, Boolean
 statusPC, Integer vermogenPC)



Bijlage 6 Commando

